Region-Based Value Iteration for Partially Observable Markov Decision Processes

Hui Li, Xuejun Liao, and Lawrence Carin
{hli1,xjliao,lcarin}@ee.duke.edu

Department of Electrical and Computer Engineering
Duke University
Durham, NC 27708-0291, USA
Outline

- Review of POMDPs
- Region-based Value Iteration (RBVI)
  - RBVI: Overview
  - RBVI: Formal Formulation
  - RBVI: Parameter Estimation
- Experimental Results and Time Complexity Analysis
- Conclusions
Review of POMDPs

The POMDP is a tuple \((S, A, O, T, \Omega, R)\)

- \(S, A, O\) — states, actions, observations
- \(T\) — action-dependent state transition matrix
- \(\Omega\) — action-dependent observation function
- \(R\) — reward function

- Stochastic state transitions — the consequence of an action is uncertain
- Partially observable states — the states are not observed directly but are inferred from perception that partially characterizes the states
- Delayed reward — the present action has influences into the future
Review of POMDPs (cont’d)

- Belief state — the belief state is a sufficient statistic that summarize the history
- Belief state MDP — the POMDP is a discrete time and continuous state MDP
- Belief update (transition)

\[
b'(s') = \frac{\sum_s b(s) T_{ss'} \Omega_{s'o}^a}{p(o|b, a)}
\]

(1)

\[
p(o|b, a) = \sum_{s'} \sum_s b(s) T_{ss'} \Omega_{s'o}^a
\]

(2)

- Piecewise linear convexity — the optimal value of an POMDP is a piecewise linear and convex function of belief state for a finite horizon
Bellman Equation

- Optimal value function $V^n(b)$ for horizon $n$

$$V^n(b) = \max_{a \in A} \left[ \sum_{s \in S} b(s) R(s, a) + \gamma \sum_{o \in O} \sum_{s', s'' \in S} b(s) T_{ss'}^a \Omega_{s', o}^{n-1} V^{n-1}(b') \right]$$  \hspace{1cm} (3)

- Piecewise linear convexity

$$V^n(b) = \max_k \left[ \sum_{s \in S} \alpha_k^n(s) b(s) \right] = \max_k (b^T \cdot \alpha_k^n)$$  \hspace{1cm} (4)

- Substituting (4) and (1) into (3),

$$V^n(b) = \max_{a \in A} \sum_{s \in S} b(s) \left[ R(s, a) + \gamma \sum_{o \in O} \sum_{s' \in S} T_{ss'}^a \Omega_{s', o}^{n-1} l(b, a, o)(s') \right]$$  \hspace{1cm} (5)

$$l(b, a, o) = \arg \max_k \left\{ \sum_{s \in S} b(s) \sum_{s' \in S} T_{ss'}^a \Omega_{s', o}^{n-1} \right\}$$  \hspace{1cm} (6)
RBVI probabilistically partitions the belief simplex into a number of ellipsoidal regions, in each of which the optimal value function $V(b)$ is linear in $b$.

Figure 1: Approximating the polyhedral piecewise linear value function by an ellipsoidal piecewise linear function
RBVI versus PBVI

Horizon $n$  

Horizon $n-1$

PBVI

$\alpha$-vectors  

Bellman Equation  

$\alpha$-vectors

RBVI

values of belief points  

Bellman Equation  

$\alpha$-vectors

Piecewise linear regressor

Figure 2: An illustration of Region-Based Value Iteration (RBVI)
Finite set of Belief Points $\mathcal{B}$

At each expansion, consider all actions $a \in A$, and draw observations by
\[ o \sim p(o | b, a), \quad \forall a \in A \]

Figure 3: An illustration of how to generate a finite set of belief points $\mathcal{B}$
Characteristics of RBVI

- RBVI maintains an $\alpha$-vector for each convex region over which the optimal value function is linear.
- RBVI jointly estimates the $\alpha$-vectors for all convex regions based on all available belief points.
- Each polyhedral convex region is approximated as one or more ellipsoidal regions.
- RBVI estimates the $\alpha$-vectors along with the position and shape of each ellipsoidal region via efficient expectation maximization (EM) and variational Bayesian EM (VBEM).
Mathematical Formulation

- $V(b)$ is modeled as

$$V(b) = \begin{cases} 
\alpha_1^T b + e_1 & \text{if } b \in B_1 \\
\alpha_2^T b + e_2 & \text{if } b \in B_2 \\
\cdots & \cdots \\
\alpha_K^T b + e_K & \text{if } b \in B_K 
\end{cases}, \quad e_k \sim \mathcal{N}(0, \mu^2_k) \quad (7)$$

- Letting $z(b) = k$ indicate that $b \in B_k$, (7) is written as

$$p(V(b) | b, z(b) = k; \alpha) = \frac{1}{\sqrt{2\pi \mu^2_k}} \exp \left\{ -\frac{(V(b) - \alpha_k^T b)^2}{2\mu_k^2} \right\} \quad (8)$$
The $k$-th ellipsoidal region is represented probabilistically as

$$p(b | z(b) = k; c, D) = \frac{1}{\sqrt{(2\pi)^d \det D_k}} e^{-\frac{1}{2} (b-c_k)^T D_k^{-1} (b-c_k)}$$ \hspace{1cm} (9)$$

where $c_k$ and $D_k$ represents the position and shape of the $k$-th region, respectively.

For any given $b$, the prior distribution of $z(b)$ over the $K$ ellipsoidal regions is assumed

$$p(z(b) = k) = \omega_k$$ \hspace{1cm} (10)$$

The joint probability of $V(b)$ and $b$ is given by

$$p(V(b), b ; \omega, c, D, \alpha) = \sum_{k=1}^{K} \omega_k p(b | z(b) = k; c, D) p(V(b) | b, z(b) = k; \alpha)$$ \hspace{1cm} (11)$$
Expectation Maximization (EM)

Given a set of belief points $\mathcal{B} = \{b_1, \cdots, b_{|\mathcal{B}|}\}$ and the associated optimal values $\mathcal{V} = \{V_1, \cdots, V_{|\mathcal{B}|}\}$, obtained by plugging $\alpha$ vectors into the Bellman equation,

- the likelihood function is given by

$$p(\mathcal{V}, \mathcal{B}; \omega, c, D, \alpha)$$

$$= \prod_{i=1}^{|\mathcal{B}|} \sum_{k=1}^{K} \omega_k p(b_i | z_i = k; c, D) p(V_i | b_i, z_i = k; \alpha)$$

(12)

- the parameters $\{\omega, c, D, \alpha\}$ that maximize the likelihood function are found by performing the EM iterations.
The EM algorithm is an iterative computation of

- **E-Step**: calculate the posterior of $z$

$$
\delta_k^i = p(z_i = k | V_i, b_i) = \frac{\omega_k p(b_i | z_i = k; c, D) p(V_i | b_i, z_i = k; \alpha)}{\sum_{k=1}^K p(b_i | z_i = k; \omega, c, D) p(V_i | b_i, z_i = k; \alpha)}
$$  \hspace{1cm} (13)

- **M-Step**: re-estimate the parameters $\{c, D, \alpha, \omega\}$

$$
\hat{\omega}_k = \frac{\sum_{i=1}^{|B|} \delta_k^i}{|B|}, \hspace{1cm} \hat{c}_k = \frac{\sum_{i=1}^{|B|} \delta_k^i b_i}{\sum_{i=1}^{|B|} \delta_k^i}
$$  \hspace{1cm} (14)

$$
\hat{D}_k = \frac{\sum_{i=1}^{|B|} \delta_k^i (b_i - \hat{c}_k)(b_i - \hat{c}_k)^T}{\sum_{i=1}^{|B|} \delta_k^i}
$$  \hspace{1cm} (15)

$$
\hat{\alpha}_k = \left[ \sum_{i=1}^{|B|} \delta_k^i b_i b_i^T \right]^{-1} \sum_{i=1}^{|B|} \delta_k^i V_i b_i
$$  \hspace{1cm} (16)
Variational Bayesian EM

Compared with EM, variational Bayesian EM

- is less sensitive to local maxima or singularities of the likelihood function
- can automatically select $K$ the number of regions
- treats $\omega$ as model hyper-parameters and find the $\omega$ that maximizes the marginal likelihood

\[
p(\mathcal{V}, \mathcal{B}|\omega) = \int p(\mathcal{V}, \mathcal{B}|\Theta, \omega)p(\Theta)d\Theta \tag{17}
\]

where $\Theta = \{c, D, \alpha\}$ and $p(\Theta)$ is the prior distribution of $\Theta$. 
The VB-EM algorithm is an iterative computation of

- **VBE-Step:** calculate the variational posterior $q(z)$

$$
q(z_i = k) \propto \exp \left\{ \frac{1}{2} \left[ \sum_{j=1}^d \psi \left( \frac{n_k^{+1} - j}{2} \right) + \ln |S_k| \right] - \frac{1}{2} \ln 2\pi \mu_k^2 + \beta_k - \frac{1}{2\mu_k^2} \left[ (V_i - \eta_k^T b_i)^2 + b_i^T \Sigma_k b_i \right] - \frac{d}{2} \left[ (b_i - m_k)^T n_k S_k (b_i - m_k) \right] \right\}
$$

- **VBM-Step:** calculate the variational posterior $q(\Theta)$

$$
\beta_k^{-1,\text{new}} = \beta_k^{-1,0} + \sum_{i=1}^{\text{|B|}} \delta_k^i, \quad m_k^{\text{new}} = \frac{\beta_k^{-1,0} m_k^0 + \sum_{i=1}^{\text{|B|}} \delta_k^i b_i}{\beta_k^{-1,0} + \sum_{i=1}^{\text{|B|}} \delta_k^i b_i} \\
n_k^{\text{new}} = n_k^0 + \sum_{i=1}^{\text{|B|}} \delta_k^i, \quad \Sigma_k^{-1,\text{new}} = \Sigma_k^{-1,0} + \sum_{i=1}^{\text{|B|}} \delta_k^i b_i b_i^T \mu_k^2 \\
\eta_k = \left( \Sigma_k^{-1,\text{new}} \right)^{-1} \left( \Sigma_k^{-1,0} \eta_k^0 + \frac{\sum_{i=1}^{\text{|B|}} b_i V_i \delta_k^i}{\mu_k^2} \right) \\
S_k^{-1,\text{new}} = S_k^{-1,0} + \sum_{i=1}^{\text{|B|}} \delta_k^i b_i b_i^T + \beta_k^{-1,0} m_k^0 m_k^0, T - m_k^{\text{new}} m_k^{\text{new}, T} \beta_k^{-1,\text{new}}
$$

- **Model selection:** re-estimate the model hyper-parameters

$$\omega_k = \frac{1}{|B|} \sum_{i=1}^{\text{|B|}} q(z_i = k)$$

Hui Li, Xuejun Liao, and Lawrence Carin

RBVI for POMDPs

ICML 2006 – p. 15/22
Table 1: The complete RBVI algorithm with an EM estimator

<table>
<thead>
<tr>
<th>Input: POMDP model ((S, A, T, O, \Omega, R)), (K), initial (\alpha)-vectors (\alpha^0), convergence criterion for the (\ell) function, convergence criterion for the value function or maximum length of horizon (T);</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output: the (\alpha)-vectors (\alpha^n)</td>
</tr>
</tbody>
</table>

1. Determine the set of belief points \(\mathcal{B} = \mathcal{B}_{ske} \cup \mathcal{B}_{exp} = \{b_1, b_2, \cdots, b_{|\mathcal{B}|}\}\); let \(n = 1\).
2. Repeat until \(n = T\) or \(\frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} V^n(b_i)\) converges in \(n\):
   2.1. Compute \(V^n(b), \forall b \in \mathcal{B}\), using (3); record the values \(\mathcal{V} = \{V_1, V_2, \cdots, V_{|\mathcal{B}|}\}\), where \(V_i = V^n(b_i), i = 1, 2, \cdots, |\mathcal{B}|\).
   2.2. Initialize \(\{\omega, c, D, \alpha\}\); let \(\mu_k = std(\mathcal{V}), k = 1, 2, \ldots, K\).
2.3. Repeat until the \(\ell\) function in (12) converges:
   2.3.1. E-Step: compute \(\delta^i_k\) using (13) along with (8) and (9).
   2.3.2. M-Step: compute the updated parameters \(\{\omega, \hat{c}, \hat{D}, \hat{\alpha}\}\) using (14), (15), and (16); let \(c = \hat{c}, D = \hat{D}, \text{ and } \alpha = \hat{\alpha}\).
2.4. Record \(\alpha^n = \alpha\); let \(n = n + 1\).
Experimental Results

Four benchmark problems considered:

- Tiger-grid, Hallway, and Hallway2 (Littman et al., 1995)
- Tag (Pineau et al., 2003)

The algorithms being compared:

- Grid (Brafman, 1997)
- PBUA (Poon, 2001)
- PBVI (Pineau et al., 2003)
- BPI (Poupart & Boutilier, 2003)
- Perseus (Spaan & Vlassis, 2004)
- HSVI (Smith & Simmons, 2004; Smith & Simmons, 2005)

- The proposed algorithm
### Experimental Results (cont’d)

Table 2: Results on the benchmark problems, where $T$ denotes time in seconds, $|\Gamma|$=number of $\alpha$-vectors, n.a.=not applicable, and n.v.=not available. The results marked with * are those we obtained by coding the respective algorithms in Matlab; other results are cited from the literature and may have been coded in languages other than Matlab and executed on computer platforms different from ours.

| Method       | Reward | $T(s)$ | $|\Gamma|$ | Method       | Reward | $T(s)$ | $|\Gamma|$ |
|--------------|--------|--------|-----------|--------------|--------|--------|-----------|
| Tiger-Grid   | $|S| = 33, |A| = 5, |O| = 17 | Honey-Grid  | $|S| = 57, |A| = 5, |O| = 21 |
| Grid         | 0.94   | n.v.   | 174       | Grid         | n.v.   | n.v.   | n.v.       |
| PBUA         | 2.30   | 12116  | 660       | PBUA         | 0.53   | 450    | 300       |
| PBVI         | 2.25   | 3448   | 470       | PBVI         | 0.53   | 288    | 86        |
| PBVI (*)     | 2.23   | 2239   | 970       | PBVI (*)     | 0.54   | 1166   | 408       |
| BPI          | 2.22   | 1000   | 120       | BPI          | 0.51   | 185    | 43        |
| Perseus      | 2.34   | 104    | 134       | Perseus      | 0.51   | 35     | 55        |
| HSVI1        | 2.35   | 10341  | 4860      | HSVI1        | 0.52   | 10836  | 1341      |
| HSVI2        | 2.30   | 52     | 1003      | HSVI2        | 0.52   | 2.4    | 147       |
| RBVI-EM (*)  | 1.95   | 135    | 10        | RBVI-EM (*)  | 0.54   | 88     | 10        |
| RBVI-VBEM (*)| 2.05   | 64     | 10        | RBVI-VBEM (*)| 0.54   | 84     | 10        |
Table 3: Results on the benchmark problems, where $T$ denotes time in seconds, $|\Gamma|$=number of $\alpha$-vectors, n.a.=not applicable, and n.v.=not available. The results marked with * are those we obtained by coding the respective algorithms in Matlab; other results are cited from the literature and may have been coded in languages other than Matlab and executed on computer platforms different from ours.

| Method       | Reward $T(s)$ | $|\Gamma|$ |
|--------------|---------------|-----------|
| **Hallway2** | $|S| = 89$, $|A| = 5$, $|O| = 17$ |           |
| PBUA         | 0.35          | 27898     | 1840     |
| PBVI         | 0.34          | 360       | 95       |
| PBVI (*)     | 0.35          | 2345      | 572      |
| BPI          | 0.32          | 790       | 60       |
| Perseus      | 0.35          | 10        | 56       |
| HSVI1        | 0.35          | 10010     | 1571     |
| HSVI2        | 0.35          | 1.5       | 114      |
| RBVI-EM (*)  | 0.30          | 90        | 15       |
| RBVI-VBEM (*)| 0.31          | 103       | 15       |
| **Tag**      | $|S| = 870$, $|A| = 5$, $|O| = 30$ |           |
| PBUA         | n.v.          | n.v.      | n.v.     |
| PBVI         | -9.180        | 180880    | 1334     |
| PBVI (*)     | n.v.          | n.v.      | n.v.     |
| BPI          | -6.65         | 250       | 17       |
| Perseus      | -6.17         | 1670      | 280      |
| HSVI1        | -6.37         | 10113     | 1657     |
| HSVI2        | -6.36         | 24        | 415      |
| RBVI-EM (*)  | -6.56         | 2481      | 15       |
| RBVI-VBEM (*)| -6.34         | 2430      | 15       |
Figure 4: The $\omega_k = p(z = k)$ estimated by VBEM.
Time Analysis in Big O Notation

Table 4: Time Comparison in Big O Notation, where \( \dim(\tilde{b}) \) denotes the dimensionality of \( \tilde{b} = U^T b \) and \( U \) is a matrix whose orthonormal columns span the space of \( \tilde{b} \); \( N_\Upsilon \) denotes the number of EM or VBEM iterations.

<table>
<thead>
<tr>
<th>Computation with Bellman equation</th>
<th>EM or VBEM computation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBVI ( O(</td>
<td>B</td>
</tr>
<tr>
<td>RBVI ( O(</td>
<td>B</td>
</tr>
</tbody>
</table>

The second stage of RBVI can be computed more efficiently than PBVI when: either (i) \( \frac{|S|}{|A||O|} \) is not too large, as in Tiger-Grid, Hallway, and Hallway2; or (ii) \( \dim(\tilde{b}) = \dim(U^T b) \ll \dim(b) = |S| \) as in Tag where \( \dim(\tilde{b}) = 176 \).
Conclusions

- An approximate value iteration algorithm, region-based value iteration (RBVI), is proposed to find the optimal policy for the POMDP.

- The RBVI maintains a single $\alpha$-vector for each ellipsoidal belief region over which the optimal value $V^n(b)$ is linear in $b$ for finite horizon $n$.

- The compact representation of the value function brings significant computational savings.

- The results on benchmark problems show the RBVI produces performance that is competitive to state-of-the-art POMDP algorithms by using a significantly smaller number of $\alpha$-vectors.